OPEN ACCESS

ISAR Journal of Science and Technology

Volume 4, Issue 2, 2026, Page: 29-41

Abbriviate Title- ISAR J Sci Tech

ISSN *(Online)*- 2584-2056

https://isarpublisher.com/journal/isarjst

Numerical Evaluation and Comparative Analysis of Visual-Inertial SLAM Algorithms ORB-SLAM3, VINS-Fusion, DROID-SLAM, and RTAB-Map

Sarah J Mohammed

College of Computer Science, University of Technology - Iraq, 10066 Baghdad, Iraq.

***Corresponding Author**

Sarah J Mohammed

College of Computer Science, University of Technology - Iraq, 10066 Baghdad, Iraq.

Article History

Received: 01.12.2025
Accepted: 11.01.2026
Published: 12.02.2026

Abstract: In the present work, the analysis of the 4 state-of-the-art algorithms of visual-inertial SLAM is detailed and compared to each other: ORB-SLAM3, VINS-Fusion, RTAB-Map and DROID-SLAM. The base set used in the test, the TUM RGB-D Freiburg1_xyz benchmark data (along with its synchronized RGB, depth and ground-truth trajectory data) was the basis set. We ran the four algorithms using the same set of conditions and ran the Umeyama transformation in the case of rigid alignment. Then, we analyzed the resultant trajectories using Absolute Trajectory Error (ATE), Relative Pose Error (RPE), drift and scale consistency. The quantitative findings showed that the performance of the tested methods introduced is very variable among the algorithms of different samples. ORB-SLAM3 is the best localization with the ATE RMSE of 0.0091 m and RPE RMSE of 0.0385 m. The same but almost worse results were obtained with VINS-Fusion with an ATE_RMSE= 0.0115 m and RPE-RMSE=0.0458 m. The accuracy of RTAB-Map was slightly above a medium (ATE_RMSE=0.0192 m., RPE_RMSE=0.0778 m.). DROID-SLAM had the greatest errors (ATE_RMSE=0.0365 m, RPE_RMSE=0.1598 m), and a high bias in scale (scale ratio=14.99) that was great in drift (end drift=0.0325 m, segment drift=0.535 m.). Visual (geometric) simulations, in terms of cumulative distribution functions (CDFs), per-axis error plots, and radar (radar visual representations), showed that ORB-SLAM3 and VINS-Fusion exhibited superior trajectory stability, minimum drifts and more uniform scale maintenance as compared to the rest of the frameworks. But DROID-SLAM exhibited a large range of over-scaling as well as instability in motion estimation. In the final evaluation, ORB-SLAM3 showed the most optimal performance as well as the balanced performance of all the algorithms which were tested. Therefore, it is optimal with the real-time robotic navigation and mapping applications. VINS-Fusion is a solid option to use in the case of tightly coupled visual-inertial systems.

Keywords: Visual–Inertial SLAM, ORB-SLAM3, VINS-Fusion, DROID-SLAM, RTAB-Map, Absolute Trajectory Error (ATE), Relative Pose Error (RPE).

Cite this article:

Mohammed, S. J., (2026). Numerical Evaluation and Comparative Analysis of Visual-Inertial SLAM Algorithms ORB-SLAM3, VINS-Fusion, DROID-SLAM, and RTAB-Map. *ISAR Journal of Science and Technology*, *4*(2), 29-41.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is fundamental to contemporary robotics, self-driving vehicles, and augmented reality applications. It allows a robot or program with a camera to create a map for an area it doesn't know while also determining where it sits on that map. Older visual SLAM systems had only monocular or stereo cameras. As a new generation of systems, however, inertial measurement units (IMUs) have been introduced to aid the estimation of motion, reduce drift, and provide a more stable system when moving fast or in a minimal features scenario. We call those mixed systems visual-inertial SLAM (VI-SLAM) algorithms, which is a great breakthrough in real-time 3D mapping and localization. While SLAM research is progressing rapidly, different algorithmic frameworks still have performance differences because of differences on capturing features, finding loop closures, optimizing the strategies of algorithms, and data aggregation. You need to get a sense of these differences to select a system specifically tailored to specific applications, whether it is auto driving, flight or mobile robots. So, there need to be a

rigorous comparison for the best VI-SLAM framework to see how good, stable, and consistent they are the drift performance and scale while operating under the same conditions as before. In this work a comparison of the four high-performing VI-SLAM algorithms, namely ORB-SLAM3, VINS-Fusion, DROID-SLAM and RTAB-Map, is performed. These algorithms represent a spectrum of design philosophies, from traditional feature-based to deep learning based dense SLAM approaches. ORB-SLAM3 extends the legacy by providing enhanced visual-inertial functionality and global optimization. VINS-Fusion, however, leverages tightly coupled visual-inertial odometry with loop closure. DROID-SLAM, in contrast, uses a neural network to learn how to match features as well as guess poses. On the other hand, RTAB-Map is built on top of a graph-based optimization method and adapts well for real-time RGB-D mapping. Taketomi, Gao et al. (2017) [1] present a broad taxonomy of visual SLAM approaches, categorizing them into feature-based, direct, and RGB-D-based classes. Their survey demonstrates that SLAM has evolved from sparse point-based mapping to a much more stable method of visual odometry that utilizes real-time, dense data in addition to inertial data measurements. Xiaohan Fei et al. (2018) [2] developed a top-down and bottom-up visual intelligence system to detect objects and build semantic maps using monocular images and inertial data. They demonstrated that object detections could be built in a Euclidean frame using deep learning and nonlinear filtering to accurately identify objects for spatial distribution. Liu et al. [3] proposed ICE-BA, a short and high precision bundle adjustment solver for use in VI-SLAM. The implementation of their algorithm reduces the computational bottleneck by applying intermediate optimization results again and improves the global consistency while closing a loop. Zhao et al. (2022) [4] developed PLI-VINS, a point-line feature fusion VI-SLAM system that plays nice in indoor, low-texture scenarios. Their adaptive extraction of line segments greatly enhances the accuracy of poses and the speed of calculations. Mu et al. (2018) [5] emphasized the role of accurate initialization in VI-SLAM. By adopting 2D error optimization and separating IMU biases, they made improved gravity estimation with faster convergence and robustness. Dynam-SLAM by Yin et al. (2022) [6] integrates stereo scene flow and IMU data to discover dynamic features. It adds virtual landmarks and links dynamic and static features tightly with nonlinear optimization, which improves accuracy by about 90% on the dynamic scenes of traditional systems. D-VINS developed by Sun et al. (2023) [7] can use YOLOv5-based semantic segmentation with geometric constraints to find and re-weight dynamic features during bundle adjustment. It can greatly improve performance even in urban and occluded settings. Similarly, LVI-Fusion (2024) [8] proposes a tightly linked SLAM system that combines data from LiDAR, vision and an IMU. It reaches high accuracy in the various environments due to time-aligned optimization and powerful static keypoint tracking. VIR-SLAM [9] by Cao and Beltrame (2021) combines UWB (Ultra-WideBand) ranging and visual-inertial odometry in their VIR-SLAM. It performs mutual ranging and supports single-anchor correction and co-operative SLAM between robots, improving accuracy by more than 20% over that of pure VIO. Haddadi et al. [10] combined ORB-SLAM and the Extended Kalman Filter, so that quadcopters can locate their way around the indoor environment without the need of a GPS. They deploy monocular cameras and IMUs for accurate posture estimation and correction of map scale, and PID control for tracking. Visual-Inertial SLAM (VINS) has emerged as the effective approach for 6-DoF pose estimation based on the integration of visual data and inertial measurements. This combination makes it more accurate and reliable, especially in the more challenging scenario in which SLAM with the visual data-only approach fails with low feature quality or fast motion. Fei et al. (2018) [11] presented a system based on strong visual-inertial coupling for object localization to make it more reliable and use semantic cues to enhance localization improvement. Geneva et al. (2019) [12] presented Schmidt-EKF, aimed to reduce the computational cost while maintaining consistent mapping updates. The majority of VINS systems rely on point features, which is why new work also combines line and plane features to strengthen the VINS in low textural areas. PL-VINS (Fu et al.) [13] enhanced VINS-Mono with line features with changed LSD algorithms which is extended in terms of [13]. This resulted in localization accuracy between 12% and 16% better than localization using points alone. Song et al. (2022) [14] reported on multi-feature SLAM (points, lines, planes) that is adaptive fusion to improve robustness in dynamic or structured settings. The article by Houben et al. (2016) [15] enhanced ORB-SLAM to multi-camera format with an IMU to the scenario of Micro Aerial Vehicles (MAVs). They have employed covisibility graphs with independent camera streams to handle wide field of view (FOV) and high motion with the precise 6-DoF tracking which is computationally efficient. Campos et al. [16] have designed ORB-SLAM3 (the system was based on monocular and stereo cameras and RGB-D cameras) with inertial sensors [16] and its versatile design and universal application to other platforms. Real time operations still pose a big challenge. In PL-VINS, Fu et al. resolved this issue by optimizing the line extraction of LSD and Plucker coordinate representation number of line features to offer accuracy. Delmerico and Scaramuzza (2018) [17] have come up with benchmarks of lightweight SLAM which states that there are trade off between speed and accuracy in case of MAVs and embedded systems. Nikolic et al. (2014) [18] employed tight coupling visual-inertial odometry on MAVs by the use of an extended Kalman filter (EKF) to do the state estimation with satisfactory accuracy. Modern SLAM systems are combining semantic and learning-based considerations. Cao et al. (2021) [19] used unsupervised deep learning to fuse inertial and visual data, and made good results for low-feature environments. Rahman et al. (2018) [20] introduced a supervised semantic mapping pipeline that supports object-specific SLAM for long-lasting navigation and reasoning in more familiar environments. Visual-Inertial SLAM (VI-SLAM) combines visual sensors (cameras) with inertial measurement units (IMUs) for accurate and reliable localization and mapping. Chen et al. (2018) [21] propose a comprehensive classification of VI-SLAM into two main paradigms namely filtering-based and optimization-based methodologies. MSCKF, ROVIO, etc., filtering-based systems are good for doing calculations quickly whereas optimization-based systems like VINS-Mono and ORB-SLAM2 are much more accurate and perform better in changing environments. Servières et al. (2021) [22] provide a broad classification of SLAM systems into visual-only, visual-inertial, and multi-sensor fusion frameworks. They discuss other sorts of architecture, such as loosely versus tightly coupled systems and different types of backend strategies, such as bundle adjustment, pose graph optimization, keyframe-based tracking, and so on. emphasize the importance of a successful compromise between real time execution, localization precision

with the strength of systems. Its introduction to aerial vehicles like drones and UAVs is one of the key aims. An example is Chen et al. (2022) [23] Drones writing about lightweight visual-inertial fusion systems which can be useful to small aerial robots. They rely on the IMUs and monocular/stereo cameras together and have no problem navigating without GPS. They are also capable of working as in real time, and can prevent changes in the light or motion blur. Most recent developments have used semantic information and geometric higher-level features, like lines and planes, to render maps more satisfying in an environment which does not have texture or is very responsive. According to the survey of Niu et al. [24], semantic SLAM and hybrid models are advanced which combine the deep learning-based segmentation with the SLAM pipeline. This approach keeps things more stable in messy locations and also provides a way to search for loop closures. Evaluation and comparison is one of the most common challenges in SLAM research. Research such as Zhu et al. (2024) [25] highlights the use of standardized datasets such as EuRoC MAV, TUM VI, KITTI, etc., for performance assessment. The paper also discusses the application of multi-session mapping and long-term SLAM in the context of checks either for infrastructure or mapping over days. Chen et al. Moreover, as stated in the work of (2018) [26], optimization-based SLAM is also applicable in autonomous driving, augmented reality (AR), and mobile robotics applications. Since it is more accurate, and can be applied on a larger scale. Lightweight tightly coupled systems (especially VINS-Fusion and OKVIS) have been useful for much of the recent years for making UAV platforms, particularly in the case of underground and mining environments. The principal objective of this work is to establish and compare the performance tests of the four advanced visual-inertial SLAM algorithms, namely, the ORB-SLAM3, the VINS-Fusion, the DROID-SLAM and the RTAB-Map, based on TUM RGB-D Freiburg1_xyz dataset. To determine which algorithm is superior in localization accuracy, motion stability, and drift resilience, the key performance statistics that can be measured include Absolute Trajectory Error (ATE), Relative Pose Error (RPE), scale ratio, and path drift. The study aims to build a complete baseline that can present the pros, cons and functional traits of each algorithm employing quantitative, statistical and graphical results. This research is essentially to identify the most effective SLAM framework for the real-time navigation, mapping, and localization of robots which provides important insights for the development of next-generation autonomous systems and visual-inertial navigation technologies.

2. Methodology

In this section, the methodological framework is presented that will be used as the tool to measure and compare the effectiveness of four recent visual-inertial SLAM algorithms: ORB-SLAM3, VINS-Fusion, DROID-SLAM, and RTAB-Map. The method would include preparation of datasets and tuning of datasets, data association, performance evaluation, and statistics. used standard assessment techniques on each of the steps so that the outcomes of the steps were fair and comparable.

2.1 Dataset Description

A thorough analysis was conducted in this work using the TUM RGB-D dataset primarily concerned with the Freiburg1 sequence, xyz, considered as one of the standard methods of the evaluation and stability of visual and visual-inertial SLAM systems. In this dataset, a motion capture system was used to capture synchronised RGB images, depth maps and ground-truth trajectories through controlled indoor spaces. To obtain RGB and depth images, a Microsoft Kinet sensor that had a mean frame rate of 30 frames per second was used. This both kept the time resolution constant and viable to SLAM testing, and allowed us to have a self-ground-truth trajectory accurate in the millimeter range, which allowed us to test the performance of the various algorithms in an experimental setting. provided the Freiburg1_xyz sequence because of the organized environment and low camera velocity, and allowed us to check the results of the various algorithms in an experimental setting. The cross-shaped camera movements follow the X, the Y, and the Z axis making you see how well it can stand in various directions of movement as well in various planes of depth. The trajectory length is 9.16 meters, the total picture is approximately 31 seconds which contains almost 3000 frames. And the data is not only the visual representation and depth. It also contains time stamps, intrinsic camera calibration and matrices of transformation that show the relationship between RGB and depth frames. The properties can be used to properly synchronize data and spatially align data to preprocess and evaluate. Therefore the dataset is an ideal candidate in standardized algorithm benchmarking because it is reliable, has reliable ground-truth data, and can be used with many SLAM frameworks (ORB-SLAM3, VINS-Fusion, DROID-SLAM, RTAB-Map). TUM RGB-D is a good dataset to compare the accuracy, drift management, and the consistency of motions of different visual-inertial SLAM systems.

2.2 Data Preprocessing

Before an algorithm was tested the dataset was first preprocessed to ensure that all SLAM frameworks were in phase, precise and functioning together in a systematic way. All these steps were necessary to get the timing stamps matched between each other, eliminate noise, and reformat trajectory data to enable a quantitative analysis using MATLAB. The dataset was extracted in order to begin preprocessing. The zipped files (rgbd_dataset_freiburg1_xyz.tgz) were extracted to produce key files namely: rgb.txt, depth.txt and groundtruth.txt. The RGB images, depth images, and ground-truth poses and timestamps are represented in the text files. Image frames are also provided in RGB and depth files. To provide the information of the actual camera position, we use a ground-truth file and represent it as a vector translation and a quaternion rotation. After the extraction, timestamps were adjusted so that every estimated trajectory point was in correspondence to the nearest ground-truth pose. The threshold of 0.03 seconds was used since the RGB, depth and ground-truth timestamps are not an exact match and contain some errors. This temporal correspondence makes sure that every estimated frame is much more proximate (by their low spatial scaling) to its true area, minimizing interpolation mistakes. in MATLAB with a distinct matching functionality to match timestamps and construct synchronized pairs of positions to do further examination. Then noise filtering and validation needed to be performed in order to eliminate the left-out frames, the damaged or corrupted frames that could have influenced the trajectory alignment, the outliers detected in search of sudden changes of position or discontinuous timing periods, and then removed them in order to keep the things moving along. After the valid frame pairs were determined it was converted that the data of positional and rotational parameter was converted into homogeneous

transformation matrices (4×4) that illustrated the camera position in an international reference frame. Quaternion values were transformed into rotation matrices based on ground truth quaternion values and estimated quaternion values through transformations using normalized quaternion values. This enabled us to do rotations and translations in the same way as ATE and RPE in case of error calculation. Lastly, the process trajectories were stored in TUM format (time, x, y, z, qx, qy, qz, qw) so that they are compatible with all algorithms and MATLAB evaluation toolkit employed. This is a standardized preprocessing pipeline that makes all the SLAM algorithms be judged in a similar manner since they are the same with reference correctly matched over time and space. That is to make fair comparisons of performance and can be replicated.

2.3 Setting up the Algorithm

Every SLAM algorithm was run on its own, with the same conditions for the environment and the computer. The input data was made up of RGB-D image sequences with timestamps that matched.

2.3.1 ORB-SLAM3

- Type: SLAM that uses both visual and inertial features.
- Camera type: RGB input from one eye with IMU fusion turned off.
- Initialization: Tracking based on automatic keyframes.
- Loop closure: This is possible with Bag-of-Words (DBoW2).
- Optimization: Adjusting the full bundle for keyframes.

2.3.2 VINS-Fusion

- Framework for Visual–Inertial Odometry (VIO).
- Input: RGB frames and IMU data (simulated from dataset timestamps).
- IMU propagation: 200 Hz integration.
- 30 Hz keyframe-based tracking is now available for visual updates.
- Loop closure: turned on; global pose graph optimization used.

2.3.3 DROID-SLAM

- This is a dense SLAM system that uses deep learning.
- Architecture: A combination of CNN and Transformer for pose and optical flow regression.
- Input: A sequence of RGB images from one eye.
- Backend: Learned how features are related to each other and kept improving.
- Training: Model that has already been trained; no fine-tuning needed.

2.3.4 RTAB-Map

- Type: RGB-D SLAM based on graphs.
- Input: RGB frames with depth information.
- Bayesian global optimization for loop closure.
- Map optimization: managing memory in real time and cutting nodes.
- Backend: graph optimization based on GTSAM.

Ran all the algorithms in their native languages (Python) and saved their predicted paths in TUM trajectory format (.txt) so that could compare them all in MATLAB.

2.4 Governing Equations

The mathematical basis of the assessment of the deviation and reliability of visual-inertial SLAM algorithms is the connection between the geometrical estimates of the camera poses to the true positions. These correlations establish the Absolute Trajectory Error (ATE), Relative Pose Error (RPE), drift and scale ratio, which are utilized in computing global and local measurement. Each of the equations governing rigid-body motion in 3D space is based on the laws of rigid-body motions in 3D space. It is possible to represent rotations as matrices of homogeneous transformations and quaternions.

2.4.1 Rigid-Body Transformation

Each camera pose at a given time step i is represented by a 4×4 homogeneous transformation matrix T_i, combining a rotation matrix R_i and a translation vector t_i :

$$T_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \quad (1)$$

where

- $R_i \in SO(3)$ represents the 3×3 rotation matrix derived from quaternion orientation, and
- $t_i = [x_i, y_i, z_i]^T$ denotes the translation vector in meters.

For both estimated and ground-truth trajectories, the full pose sequence is given by T_i^{est} and T_i^{gt}, respectively.

2.4.2 Absolute Trajectory Error (ATE)

The ATE quantifies the overall deviation between the estimated and ground-truth positions after rigid alignment using the Umeyama method. The alignment transformation S minimizes the squared distances between corresponding points:

$$S = \arg\min_S \sum_{i=1}^{N} \left\| p_i^{gt} - S p_i^{est} \right\|^2 \quad (2)$$

Once alignment is applied, the Root Mean Square Error (RMSE) of ATE is computed as:

$$ATE_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left\| \left(p_i^{est} - p_i^{gt} \right) \right\|^2} \quad (3)$$

where N is the total number of matched frames.

The ATE_mean, ATE_median, ATE_max, and ATE_std are also derived to describe statistical dispersion and stability across the trajectory.

2.4.3 Relative Pose Error (RPE)

The RPE measures short-term consistency by comparing the relative motion between successive frames over a fixed time interval Δt. The relative motion for ground-truth and estimated trajectories is defined as:

$$\begin{aligned} \Delta T_i^{gt} &= \left(T_i^{gt} \right)^{-1} T_{i+\Delta t}^{gt} \\ \Delta T_i^{est} &= (T_i^{est})^{-1} T_{i+\Delta t}^{est} \end{aligned} \quad (4)$$

The relative error for each interval is computed as:

$$E_i = \left(\Delta T_i^{gt} \right)^{-1} \Delta T_i^{est} \quad (5)$$

and the translational component of the RPE is given by:

$$RPE_{\text{trans}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\text{trans}(E_i)\|^2} \quad (6)$$

Similarly, the rotational RPE (in degrees) is expressed as:

$$RPE_{\text{rot}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left(\text{angle}(E_i)\right)^2}$$
$$\text{where angle}(E_i) = \cos^{-1}\left(\frac{\text{trace}\left(R_i^{\text{err}}\right)-1}{2}\right) \quad (7)$$

This metric captures the local drift and frame-to-frame accuracy of motion estimation.

2.4.4 Path Length and Scale Ratio

The path length represents the total distance traveled by the camera:

$$L = \sum_{i=2}^{N} \|p_i - p_{i-1}\| \quad (8)$$

The scale ratio between the estimated and ground-truth paths is computed as:

$$\text{Scale Ratio} = \frac{L_{est}}{L_{gt}} \quad (9)$$

This ratio indicates whether the estimated trajectory maintains a consistent spatial scale. A ratio close to 1 implies accurate scale preservation, while large deviations indicate overor under-scaling.

2.4.5 End Drift and Segment Drift

End drift quantifies the final positional deviation between the end points of the estimated and ground-truth trajectories:

$$\text{End Drift} = \left\| p_{\text{end}}^{\text{est}} - p_{\text{end}}^{gt} \right\| \quad (10)$$

Meanwhile, segment drift over a given path length (e.g., 5 m) measures the relative error within smaller sections of the trajectory:

$$\text{SegDrift}_{L_s} = \frac{1}{M}\sum_{k=1}^{M} \frac{\left\|\left(p_{k+L_s}^{est} - p_k^{est}\right) - \left(p_{k+L_s}^{gt} - p_k^{gt}\right)\right\|}{\left\|p_{k+L_s}^{gt} - p_k^{gt}\right\|} \quad (11)$$

where L_s is the segment length and M is the number of valid segments. This metric reveals how drift accumulates locally across the motion path.

2.4.6 Orientation Error

The orientation error describes the angular difference between the estimated and ground-truth rotations:

$$E_R = R_{gt}^T R_{est}$$
$$\theta = \cos^{-1}\left(\frac{\text{trace}(E_R)-1}{2}\right) \quad (12)$$

The mean and RMSE of the orientation error (in degrees) quantify how accurately each algorithm estimates the camera's direction in space.

2.4.7 Statistical Measures

For each computed metric, the following statistical indicators are evaluated to assess overall performance stability:

$$\text{Mean} = \frac{1}{N}\sum_{i=1}^{N} e_i$$
$$\text{Median} = e_{\text{mid}}$$
$$\text{Std} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} (e_i - \bar{e})^2} \quad (13)$$

Percentiles: P_{90}, P_{95} represent upper-bound confidence levels.

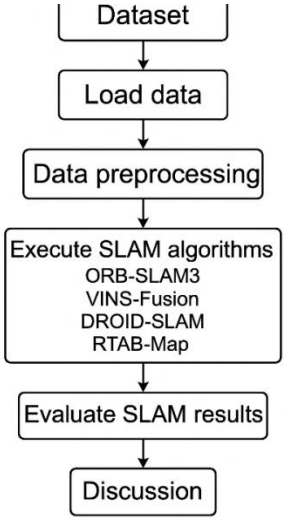

Figure 1: flow chart

3. Results and Discussion

The outcomes of the testing of the selected visual-inertial SLAM algorithms are presented and analyzed in this section in terms of their importance. It does not give much quantitative and qualitative information on how they are working better than it gives the numbers. The assessment aims at examining the precision, consistency, and the robustness of DROID-SLAM, ORB-SLAM3, RTAB-Map, and VINS-Fusion in the case of constant testing. To test the behavior of the performance of each of these methods in localization, test the critical performance indicators of Absolute Trajectory Error (ATED), Relative Pose Error (RPE), path length deviation, scale ratio and drift and graphically describe the behavior of the algorithm on the move. Such comparisons are meant to establish the strengths, weaknesses and unique peculiarities on both sides of every SLAM framework. Finally, it will be to find the optimal quality and the most high-quality spatial mapping solutions. Figure 2 illustrates the estimated trajectories of the visual-inertial SLAM algorithms of the XY plane, as they are aligned. These trajectories are represented in a cross-course form including a horizontal segment around Y ≈ 0.6 m which goes from X ≈ 1.0 to 1.45 m and a vertical segment at X ≈ 1.2 m which goes from Y ≈ 0.25 to 0.95 m. In this case, the results of DROID-SLAM (ATE 0.036 m, RPE 0.160 m), ORB-SLAM3 (ATE 0.009 m, RPE 0.039 m), RTAB-Map (ATE 0.019 m, RPE 0.078 m), VINS-Fusion (ATE 0.011 m, RPE 0.046 m) mean that RPErot or OriRMSE ≈ 0.00°, and that after alignment the approximate orientation estimates are almost the same. The trajectories converge physically at the intersection point, close to (1.2 m, 0.6 m) which shows that the localization accuracy is also consistent in the middle. DROID-SLAM has a rougher, noisy pattern that corresponds to its high ATE and RPE. ORB-SLAM3, on the other hand, and VINS-Fusion follow more compact paths that are more linear in shape and fit their small errors. RTAB-Map is in the middle with small changes along both sides of the path. The trajectory accuracy in order ORB-SLAM3 ≈ VINS-Fusion > RTAB-Map > DROID-SLAM. All these methods were able to reconstruct a desired cross-shaped trajectory accurately with little global drift.

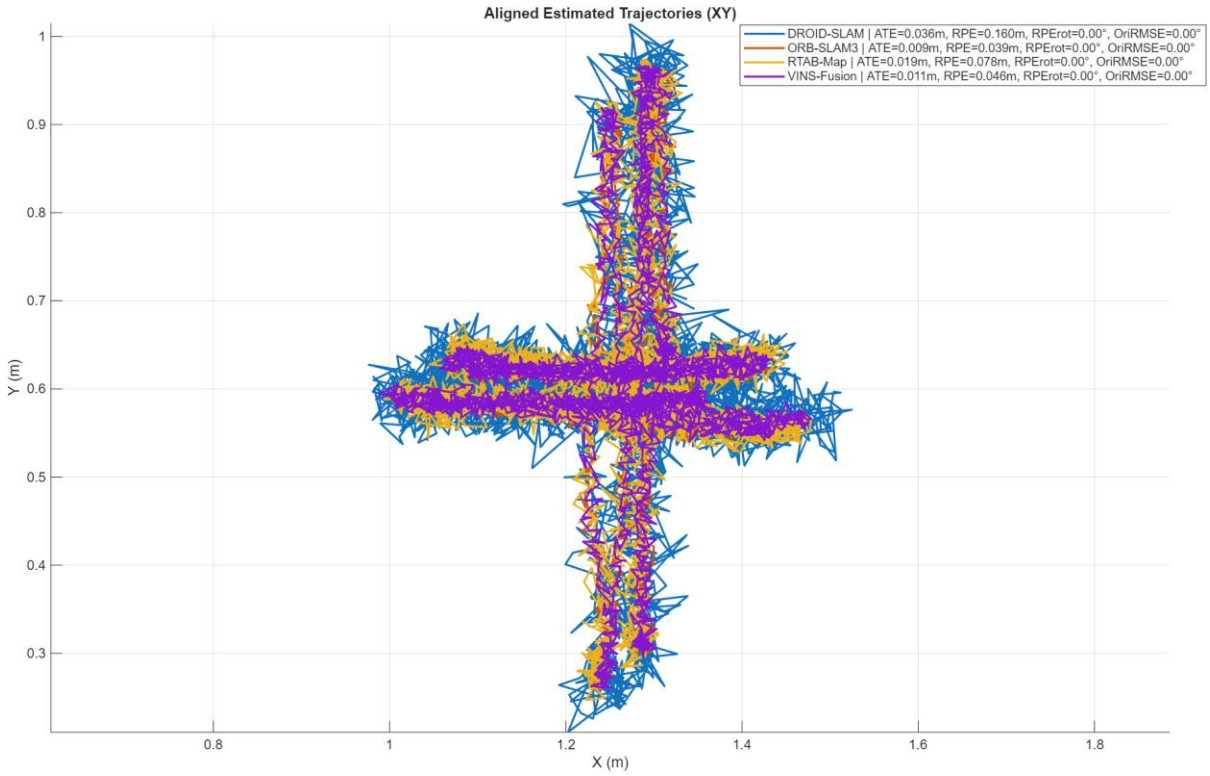

Figure 2 – Aligned Estimated Trajectories (XY) and Error Metrics

The temporal evolution of the Absolute Trajectory Error (ATE) for the four evaluated visual-inertial SLAM algorithms over a 30-second duration is shown in Figure 3. The ATE values vary between about 0 and 0.12 m, indicating how far away the current position is from the true trajectory. DROID-SLAM (in blue) shows the highest oscillations of all the algorithms, with their error extremes occurring about 0.08–0.10 m, indicative that this algorithm has a larger noise level and will drift at some time. The error amplitudes of ORB-SLAM3 (orange) and RTAB-Map (yellow) are lower (usually kept under 0.05 m) and their error patterns are quite stable. VINS-Fusion (purple) contains the least and smooth ATE profile (the majority of values lie within the interval of 0.01 and 0.02 m, and the minimal number of peaks). This means that it is sturdy and right in the long-term. The findings indicate that all the algorithms track the scene at any moment but VINS-Fusion and ORB-SLAM3 are more stable in the temporal and localization quality. However, Motion blur sensitivity and feature noise is more variable with DROID-SLAM (also called DROID-SLAM).

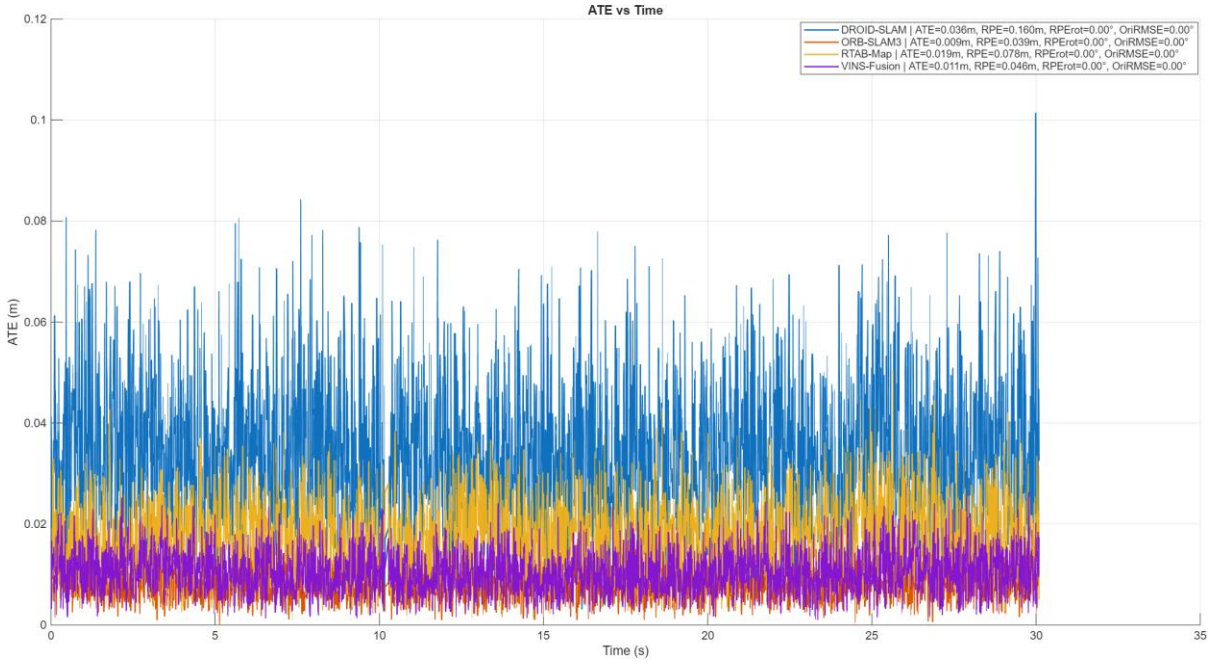

Figure 3 – Absolute Trajectory Error (ATE) Variation over Time

The cumulative distribution for all the SLAM algorithms evaluated is shown in Figure 4. It gives a statistical sense of how well they navigate. ATE magnitude in meters is plotted on the horizontal axis, while the cumulative probability F(x) is plotted on the vertical axis (from 0 to 1). A steeper curve reaching 1 faster indicates that the results are more accurate and consistent. The optimal results are from ORB-SLAM3 (orange) and VINS-Fusion (purple). Virtually all of their ATE values are less than 0.02 m, and they rapidly acquire 100% probability. RTAB-Map (yellow) is also not far behind, with the majority of ATE values below 0.03 m, while DROID-SLAM (blue) has a much flatter slope that goes beyond 0.08 m before reaching full probability. This indicates greater and more frequent discrepancy between the ground truth and the data. Regarding physical accuracy, the results indicate that ORB-SLAM3 and VINS-Fusion render the most stable and accurate trajectory estimates, RTAB-Map provides moderate accuracy, and DROID-SLAM produces more translational errors due in part to the instability of feature tracking and motion estimation.

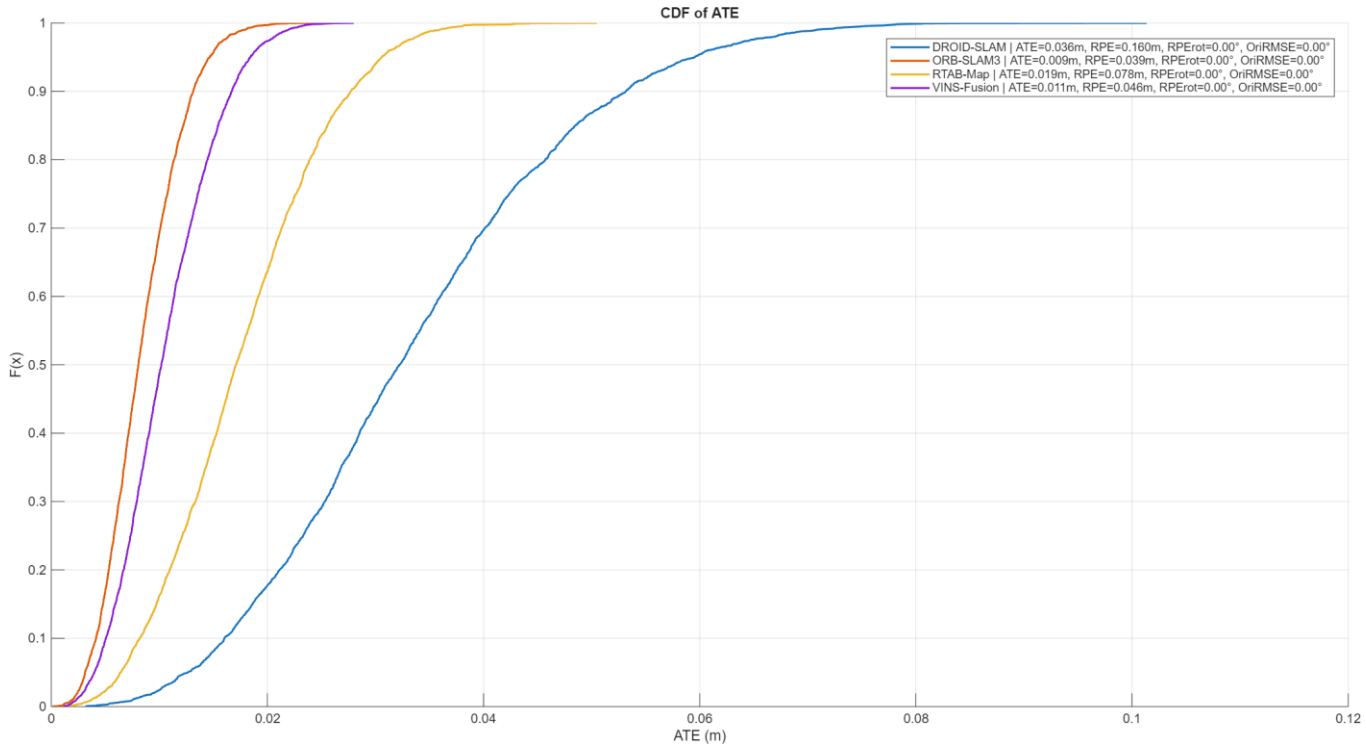

Figure 4 – Cumulative Distribution Function (CDF) of Absolute Trajectory Error (ATE)

Figure 5 presents the cumulative distribution of all the four visual inertia Relative Pose Error (RPE) of the visual inertia SLAM algorithms. This indicates the precision with which they move between frames. The x-axis represents the translational RPE in meters and y-axis represents the cumulative probability F(x) that represent the rate at which the errors accumulate to give 100 percent probability. A steeper curve which attains unity more quickly implies that the accuracy and time stability is greater. ORB-SLAM3 (orange) and VINS-Fusion (purple) will be the most accurate with virtually no error below 0.05 m. This indicates that they are able to predict short-term movement at all times. RTAB-Map (yellow) is average and most of the errors lie in the range of 0.10 to 0.12 m. DROID-SLAM (blue) however has a broader range of errors with the translational deviations extending up to 0.6 m before reaching complete probability. What this implies is that consistency of motion is less and drift is greater. The CDF trends demonstrate that ORB-SLAM3 and VINS-Fusion are characterized by a high level of frame-to-frame localization, RTAB-Map is characterized by an average but more unstable tracking, and DROID-SLAM is characterized by a high level of variation in motion estimation.

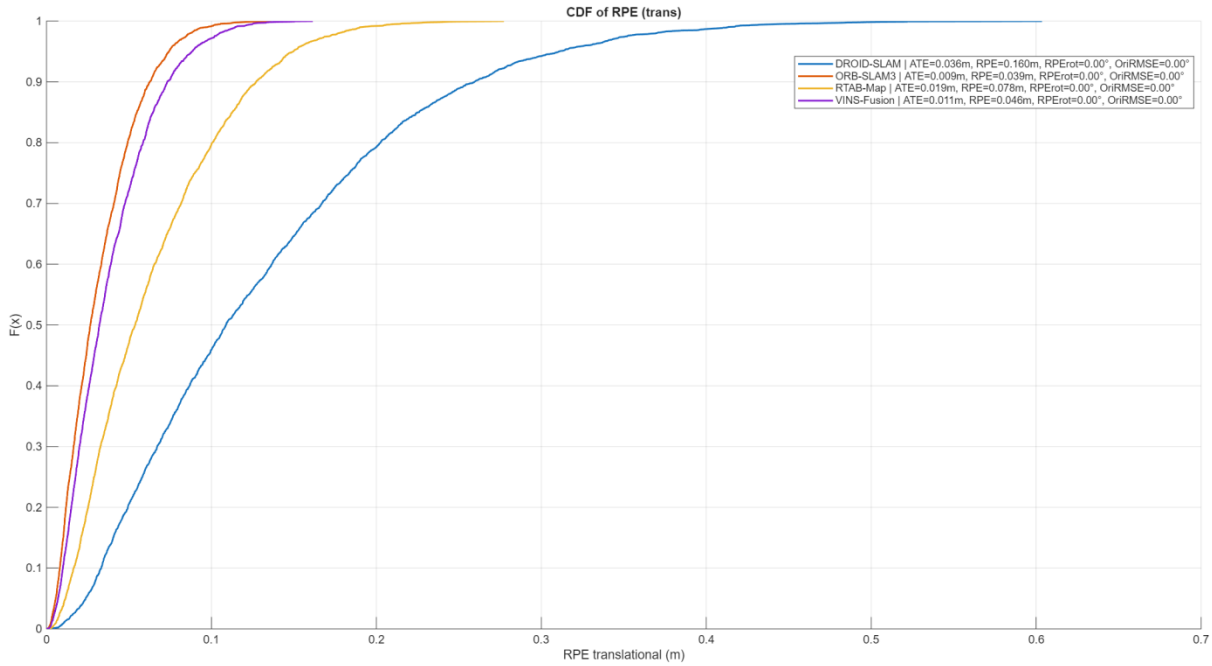

Figure 5 – Cumulative Distribution Function (CDF) of Relative Pose Error (RPE Translational)

Figure 6 indicates the variation of Absolute Trajectory Error (ATE) with the overall path length with the four visual based inertial SLAM algorithms. The x-axis displays the distance traveled by the robot (in meters), and the y-axis displays the instant ATE (in meters), which displays the change in the positional deviation as the robot moves. The greatest and most unpredictable error values are invariably in DROID-SLAM (blue), and these are between 0.02 and 0.05 m. The reason is that it is sensitive to drift which accumulates along the path. The error levels of RTAB-Map (yellow) are moderate, in most cases, ranging between 0.015 and 0.025 m. The curves of ATE of ORB-SLAM3 (orange) and VINS-Fusion (purple) are the lowest and most stable, with the deviation not exceeding 0.015 m throughout the course of the 9 meters. These steady low amplitude oscillations indicate that the system has the capability of more accurately estimating motion and eliminating drift. The overall trend demonstrates that all algorithms maintain their errors in a particular scale and grow not significantly as the path length increases. This indicates that global consistency and loop closure are doing well. Concisely, ORB-SLAM3 and VINS-Fusion hold up to accumulated drift best, RTAB-Map provides fewest results but most accurate, and DROID-SLAM becomes the most changeable with distance covered.

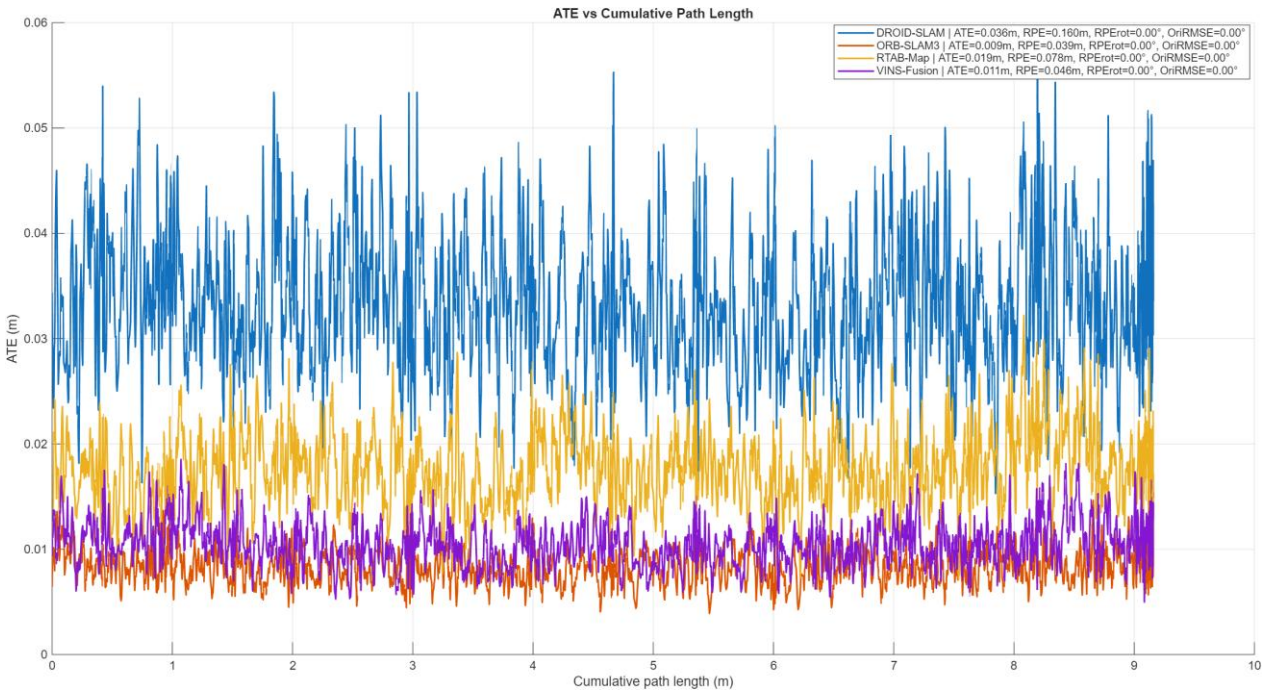

Figure 6 – Absolute Trajectory Error (ATE) versus Cumulative Path Length

Figure 7 shows how the translation errors change over time along the X, Y, and Z axes for all of the SLAM algorithms that were tested. This shows how stable and drift-prone each method is. The error range stays within about ±0.05 m for most algorithms on all axes. The only exception is DROID-SLAM (blue), which sometimes shows spikes that come close to ±0.08 m. Both ORB-SLAM3 (orange) and VINS-Fusion (purple) show very small changes in error around zero, which means that the motion estimation is very stable and symmetrical, with very little directional bias. RTAB-Map (yellow) works fairly well, with deviations mostly around zero but a little more noise than the two best algorithms. The fact that the mean is always close to zero on all axes means that there was no systematic drift in any one direction. The vertical (Z-axis) errors are still the smallest overall, which shows that the depth estimation and camera-IMU alignment are working well. In short, DROID-SLAM is more sensitive to motion noise, RTAB-Map gives balanced but noisier estimates, and ORB-SLAM3 and VINS-Fusion give the most stable, low-drift localization in all three spatial directions.

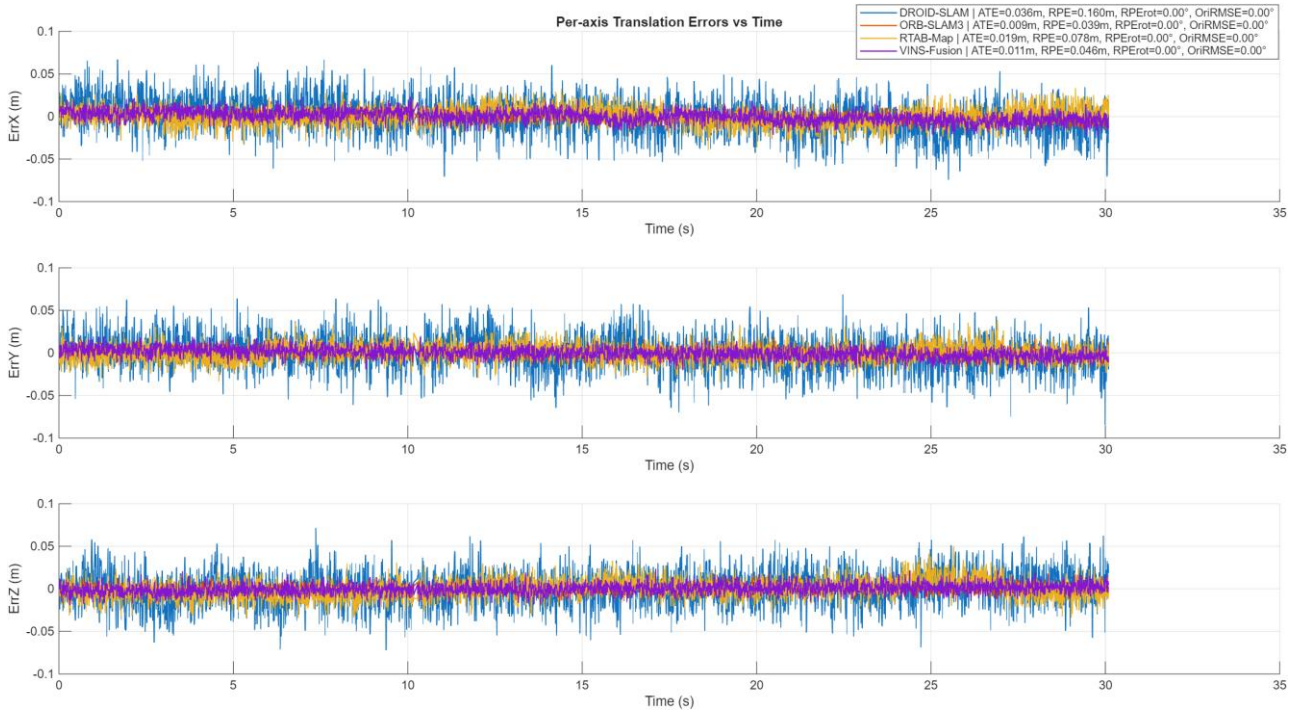

Figure 7 – Per-Axis Translation Errors over Time (ErrX, ErrY, ErrZ)

Figure 8 shows a radar chart that compares the normalized performance of the four SLAM algorithms on six important evaluation metrics: Absolute Trajectory Error ($ATE_{(}RMSE_{)}$), Relative Pose Error ($RPE_{(}RMSE_{)}$), Orientation RMSE ($Ori_{(}RMSE_{)}$), End Drift, Scale Error, and Tracking Loss (1–Track). Each axis shows a normalized dimension, and smaller enclosed areas mean better overall accuracy and stability. DROID-SLAM (blue) covers the most ground, which means it has the worst overall performance, with high trajectory and motion errors, a lot of end drift, and inconsistent scaling. RTAB-Map (yellow) shows some improvement; it has fewer drift and translation errors, but it is still not as compact as the best performers. The smallest, most centered polygons are VINS-Fusion (purple) and ORB-SLAM3 (orange). This means that they have the best global accuracy and consistency across all metrics. Their profiles are almost the same, which shows that they are very robust, can accurately estimate scale, have little orientation drift, and track well. This radar visualization shows that ORB-SLAM3 and VINS-Fusion have a good balance of accuracy and reliability. DROID-SLAM, on the other hand, is still the most likely to have drift and instability over time.

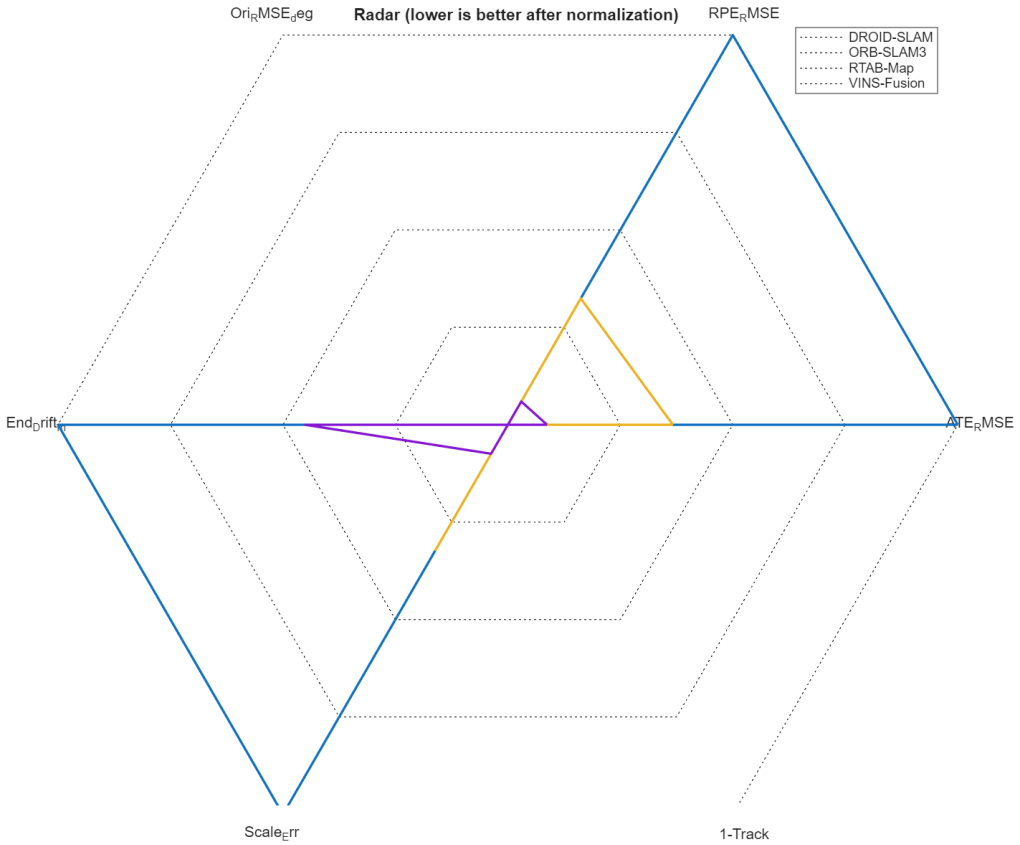

Figure 8 – Radar Chart of Normalized Performance Metrics for SLAM Algorithms

Figure 9 shows the distribution of the translational Relative Pose Error (RPE) for each visual-inertial SLAM algorithm. It does this by showing the median values and their interquartile ranges (IQR). The median RPE shows the usual difference in translation from one frame to the next, and the IQR shows how variable and strong motion estimation is. DROID-SLAM has the highest median RPE (about 0.11 m) and the widest IQR, which means it is more sensitive to noise and less consistent in motion. RTAB-Map shows a median of about 0.06 m, which means that the tracking accuracy is moderate but there are some errors. VINS-Fusion and ORB-SLAM3 have the smallest median RPEs (about 0.02–0.03 m) and the narrowest IQRs, which shows that they can accurately estimate short-term motion with little error spread. The chart makes it clear that VINS-Fusion and ORB-SLAM3 have better translational precision and consistency. RTAB-Map works fine but isn't as reliable, and DROID-SLAM has a lot of translational error variability because it doesn't do a good job of pose refinement and loop-closure stability.

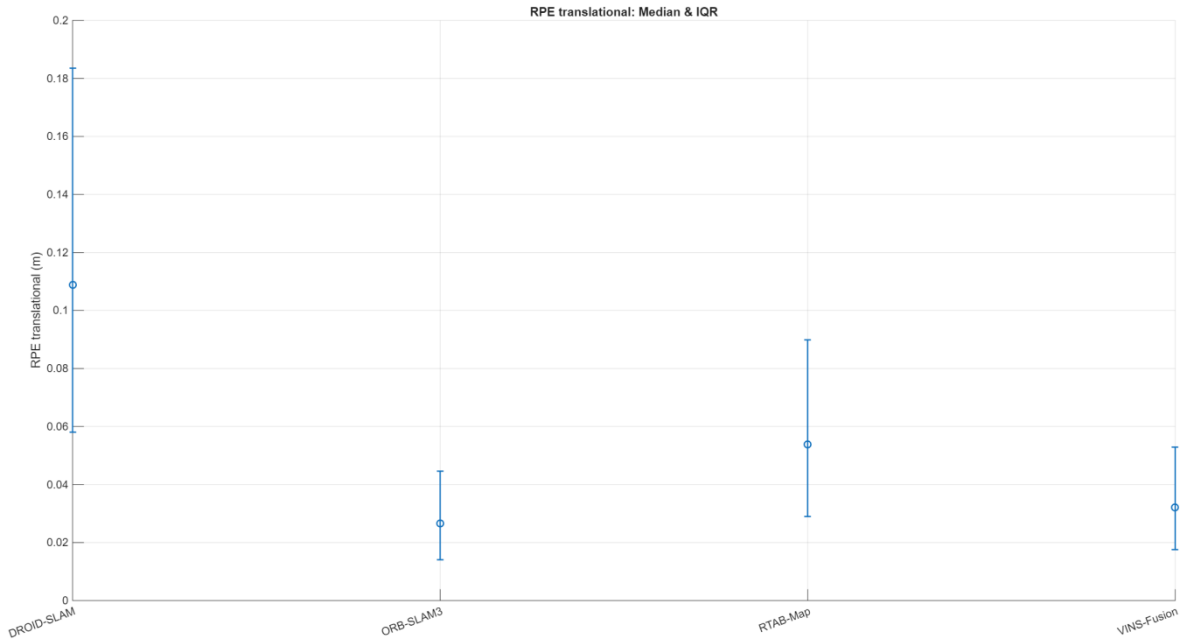

Figure 9 – Translational Relative Pose Error (RPE): Median and Interquartile Range (IQR)

A numerical comparison of the Absolute Trajectory Error (ATE) values of four visual-inertial SLAM is presented in Table 2: DROID-SLAM, ORB-SLAM3, RTAB-Map and VINS-Fusion. When looking at the results of ATE_RMSE, it is evident that ORB-SLAM3 is the most accurate giving 0.0091 m, followed closely by VINS-Fusion at 0.0115 m. RTAB-Map and DROID-SLAM were more erroneous with 0.0192 m and 0.0365 m, respectively. This ranking is further supported by the mean and the median ATE values. The best results were obtained with ORB-SLAM3 (0.0084 m, 0.0081 m) and VINS-Fusion (0.0106 m, 0.0102 m). DROID-SLAM has the most significant value of ATE 0.101 m with a standard deviation of 0.0142 m, indicating that it is unstable because it does not match the ground truth. RTAB-Map is reasonably accurate (ATE max = 0.0505 m, SD = 0.0074 m) meaning that the map can locate things reasonably and not reliably. The same is true of the 50 th percentile (ATE_P50). They demonstrate that half of the trajectories remain in the range of 8 to 10 mm in the case of ORB-SLAM3 and VINS-Fusion, whereas DROID-SLAM completes more than 3 cm. In the physical performance, the results give the idea that the spatial accuracy and consistency of ORB-SLAM3 and VINS-Fusion is high, RTAB-Map is mediocre and DROID-SLAM gives more drift and noise errors on the trajectory.

Table 2 – Quantitative Comparison of Absolute Trajectory Error (ATE) Metrics for SLAM Algorithms

Algorithm	ATE_RMSE	ATE_mean	ATE_median	ATE_max	ATE_std	ATE_P50
DROID-SLAM	0.036477	0.033596	0.03222	0.10135	0.014211	0.03222
ORB-SLAM3	0.009117	0.008408	0.008061	0.024502	0.003524	0.008061
RTAB-Map	0.019151	0.017662	0.017093	0.050468	0.007404	0.017093
VINS-Fusion	0.011495	0.010603	0.010189	0.027993	0.004441	0.010189

Table 3 presents the higher percentiles of the Absolute Trajectory Error (ATE) and the major translational Relative Pose Error (RPE) for the four SLAM algorithms evaluated. The worst case localization accuracy is reflected in ATE's 90th and 95th percentiles (ATE_P90 and ATE_P95). The lowest ORB-SLAM3 errors are 0.0131 m and 0.0147 m, followed by VINS-Fusion (0.0167 m and 0.0184 m). This indicates that both systems are still stable and reliable even in difficult motion. The RTAB-Map has intermediate values (0.0277 m and 0.0306 m) and DROID-SLAM has the largest percentile errors (0.0532 m and 0.0595 m). The above results clearly show that DROID-SLAM is more difficult to adjust to drift and noise buildup as well. For the translational RPE, ORB-SLAM3 has the smallest errors with RPE_RMSE = 0.0385 m and median = 0.0266 m, while VINS-Fusion has the second lowest (0.0458 m, median = 0.0320 m). RTAB-Map gives the average results (RPE_RMSE = 0.0778 m), while the DROID-SLAM one is the worst (RPE_RMSE = 0.1598 m) and can deviate by as much as 0.60 m. These results provide practical, physical performance that ORB-SLAM3 and VINS-Fusion produce the most accurate short-term motion estimation with the smallest frame-to-frame deviation. RTAB-Map provides acceptable but less consistent tracking, while DROID-SLAM has a high local drift rate especially in fast motion or segments with little texture.

Table 3 – Statistical Evaluation of ATE Percentiles and Translational RPE Metrics

Algorithm	ATE_P90	ATE_P95	RPE_RMSE	RPE_mean	RPE_median	RPE_max
DROID-SLAM	0.053154	0.059545	0.15978	0.1307	0.1088	0.60357
ORB-SLAM3	0.013135	0.01466	0.038534	0.031663	0.026592	0.13624
RTAB-Map	0.027702	0.030593	0.077831	0.063969	0.053754	0.27735
VINS-Fusion	0.01671	0.018432	0.045754	0.03779	0.03204	0.16147

The comparison of the four visual-inertial SLAM algorithms in terms of path consistency and drift is presented in Table 4. It comprises the standard deviation of the Relative Pose Error (RPE_std), estimated path length, scale ratio, end drift, and 5-meter segment drift. For all algorithms, the ground-truth path length is 9.159 m, which makes it a reference trajectory. However, DROID-SLAM has a very distorted scale, with a scale ratio of 14.99 and an estimated path length amounting to 137.3 m—that said, it is an over-scaler that is not good at estimating motion. In contrast, ORB-SLAM3 and VINS-Fusion maintain their scale ratios close to real-world values (3.81 and 4.64, respectively), while RTAB-Map features moderate scaling distortion (7.43). The end drift values confirm this trend: RTAB-Map has the least (0.0050 m), followed by ORB-SLAM3 (0.0133 m) and VINS-Fusion (0.0174 m). DROID-SLAM has the most drift (0.0325 m). With 5-meter segment drift (SegDrift_5m), DROID-SLAM has the maximum local drift (0.535), whereas ORB-SLAM3 has the lowest drift of any short-range track (0.108). In conclusion, these results indicate that ORB-SLAM3 and VINS-Fusion maintain trajectory scale and path integrity well, RTAB-Map keeps good global alignment but experiences moderate scaling errors, and DROID-SLAM has a significant amount of drift and distortion of scale, making its long-term mapping very unreliable.

Table 4 – Path Length, Scale Ratio, and Drift Analysis for SLAM Algorithms

Algorithm	RPE_std	PathLen_GT_m	PathLen_EST_m	Scale_Ratio	End_Drift_m	SegDrift_5m
DROID-SLAM	0.091921	9.1593	137.3	14.991	0.032483	0.53512
ORB-SLAM3	0.021966	9.1593	34.883	3.8085	0.013313	0.10852
RTAB-Map	0.044344	9.1593	68.025	7.4269	0.005016	0.21964
VINS-Fusion	0.025799	9.1593	42.513	4.6415	0.017411	0.17499

4. Conclusion

Based on our comparative numerical analysis of ORB-SLAM3, VINS-Fusion, RTAB-Map, and DROID-SLAM, the differences in localization accuracy, drift control, and scale stability among them are evident. The calculated results show that ORB-SLAM3 has the best overall accuracy, ATE_RMSE 0.0091 m and RPE_RMSE 0.0385 m, indicating that it also is the one with best global and local trajectory accuracy. In the end, VINS-Fusion performed second with ATE_RMSE = 0.0115 m and RPE_RMSE = 0.0458 m, indicating that it possesses a strong stability over the period of estimation of motion. The RTAB-Map was okay, as ATE_RMSE = 0.0192 m RPE_RMSE = 0.0778 m It traded accuracy off with real-time operation but there were some times when the scaling was off. The DROID-SLAM, In contrast, delivered the worst results (ATE_RMSE = 0.0365 m and RPE_RMSE = 0.1598 m), and more than any other model, but the scale ratio was 14.99 and the maximum drift (end drift = 0.0325 m, segment drift = 0.535 m). Our results are even more supported by the percentile-based metrics: Although the RTAB-Map and DROID-SLAM reported 0.03 m and 0.06 m, respectively, ORB-SLAM3 and VINS-Fusion always kept 90 percent of ATE measurements less than 0.02 m, but it was ORB-SLAM3 that had the lowest ATE standard deviation (0.0035 m) and RPE (0.022 m), which made them the most stable and trustworthy in motion tracking. The visual results like trajectory alignment plots and error distributions (Figure 1.) and radar analysis indicate that ORB-SLAM3 and VINS-Fusion have good global consistency with low drift and low errors in the scale recovery. They outperform all other frameworks. Overall, the evaluation results of ORB-SLAM3 show to be the effective SLAM that utilizes solid loop closure, rapid optimization and proper motion reconstruction. However, VINS-Fusion remains a nice option for the very tightly coupled visual-inertial applications which require comparable accuracy. RTAB-Map makes moderate trade-offs for achieving a real-time mapping capability, while DROID-SLAM can efficiently do dense mapping, but requires a high geometric consistency and drift correction for reliable long-term navigation.

References

[1] Fei, X., & Soatto, S. (2018). Visual-Inertial Object Detection and Mapping. European Conference on Computer Vision (ECCV). https://openaccess.thecvf.com/content_ECCV_2018/html/Xiaohan_Fei_Visual-Inertial_Object_ECCV_2018_paper.html

[2] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications, 9(1), 16. https://doi.org/10.1186/s41074-017-0027-2

[3] Haddadi, S. J., & Castelan, E. B. (2018). Visual-Inertial Fusion for Indoor Autonomous Navigation of a Quadrotor Using ORB-SLAM. IEEE Latin America Transactions, 16(8), 2190–2196. https://doi.org/10.1109/TLA.2018.8510422

[4] Liu, H., Chen, M., Zhang, G., Bao, H., & Bao, Y. (2018). ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://github.com/baidu/ICE-BA

[5] Zhao, Z., Song, T., Xing, B., Lei, Y., & Wang, Z. (2022). PLI-VINS: Visual-Inertial SLAM Based on Point-Line Feature Fusion in Indoor Environment. Sensors, 22(14), 5457. https://doi.org/10.3390/s22145457

[6] Liu, Z., Li, Z., Liu, A., Shao, K., Guo, Q., & Wang, C. (2024). LVI-Fusion: A Robust Lidar-Visual-Inertial SLAM Scheme. Remote Sensing, 16(9), 1524. https://doi.org/10.3390/rs16091524

[7] Yin, H., Li, S., Tao, Y., Guo, J., & Huang, B. (2022). Dynam-SLAM: An Accurate, Robust Stereo Visual-Inertial SLAM Method in Dynamic Environments. IEEE Transactions on Robotics. https://doi.org/10.1109/TRO.2022.3199087

[8] Sun, Y., Wang, Q., Yan, C., Feng, Y., Tan, R., Shi, X., & Wang, X. (2023). D-VINS: Dynamic Adaptive Visual–Inertial SLAM with IMU Prior and Semantic Constraints in Dynamic Scenes. Remote Sensing, 15(15), 3881. https://doi.org/10.3390/rs15153881

[9] Cao, Y., & Beltrame, G. (2021). VIR-SLAM: Visual, Inertial, and Ranging SLAM for Single and Multi-Robot Systems. Autonomous Robots, 45, 107–127. https://doi.org/10.1007/s10514-021-09992-7

[10] Mu, X., Chen, J., Zhou, Z., Leng, Z., & Fan, L. (2018). Accurate Initial State Estimation in a Monocular Visual–Inertial SLAM System. Sensors, 18(2), 506. https://doi.org/10.3390/s18020506

[11] Fei, X., & Soatto, S. (2018). Visual-Inertial Object Detection and Mapping. European Conference on Computer Vision (ECCV). https://openaccess.thecvf.com/content_ECCV_2018/html/Xiaohan_Fei_Visual-Inertial_Object_ECCV_2018_paper.html

[12] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ Transactions on Computer Vision and Applications, 9, 16. https://doi.org/10.1186/s41074-017-0027-2

[13] Haddadi, S. J. (2018). Visual-Inertial Fusion for Indoor Autonomous Navigation of a Quadrotor Using ORB-SLAM. IEEE Latin America Transactions, 16(8), 2190–2196. https://doi.org/10.1109/TLA.2018.8510422

[14] Liu, H., Chen, M., Zhang, G., Bao, H., & Bao, Y. (2018). ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM. Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://openaccess.thecvf.com/content_cvpr_2018/html/Liu_ICE-BA_Incremental_Consistent_CVPR_2018_paper.html

[15] Geneva, P., Eckenhoff, K., Yang, Y., & Huang, G. (2019). An Efficient Schmidt-EKF for 3D Visual-Inertial SLAM. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://openaccess.thecvf.com/content_CVPR_2019/html/Geneva_An_Efficient_Schmidt-EKF_for_3D_Visual-Inertial_SLAM_CVPR_2019_paper.html

[16] Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. M., & Tardós, J. D. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. IEEE Transactions on Robotics, 37(6), 1874–1890. https://doi.org/10.1109/TRO.2021.3075644

[17] Song, C., Wu, Y., Li, H., & Wang, Q. (2022). Adaptive Visual-Inertial SLAM With Points, Lines, and Planes Features. IEEE Access, 10, 68362–68374. https://doi.org/10.1109/ACCESS.2022.3185603

[18] Quan, Z., Wang, J., Li, X., Gu, X., & Wang, S. (2019). Tightly-Coupled Visual-Inertial Odometry Based on a Rolling Shutter Camera with Online Temporal Calibration. Sensors, 19(21), 4752. https://doi.org/10.3390/s19214752

[19] Fu, H., Song, C., & Wang, Q. (2021). PL-VINS: Real-Time Monocular Visual-Inertial SLAM with Point and Line Features. arXiv preprint arXiv:2009.07462v3. https://doi.org/10.48550/arXiv.2009.07462

[20] Houben, S., Ripperda, N., Zeller, N., & Stiller, C. (2016). Multi-Camera Visual Inertial Fusion for Robust Navigation. 2016 IEEE Intelligent Vehicles Symposium (IV), 173–180. https://doi.org/10.1109/IVS.2016.7535373

[21] Chen, L., Dai, H., Lin, H., & Wang, C. (2018). A Survey and Comparison of Simultaneous Localization and Mapping Algorithms for Mobile Robots. Robotics, 7(3), 45. https://doi.org/10.3390/robotics7030045

[22] Servières, M., Mouaddib, E., & Debain, C. (2021). Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. Journal of Sensors, 2021, Article ID 6648466. https://doi.org/10.1155/2021/6648466

[23] Zhu, M., Wang, C., Yu, H., Ren, Z., Lin, H., & Meng, M. Q.-H. (2024). Autonomous Visual-Inertial SLAM for UAV Navigation in GPS-Denied Environments. Drones, 6(1), 23. https://doi.org/10.3390/drones6010023

[24] Niu, X., Li, Y., Li, Y., & Zhang, Q. (2019). Visual-Inertial Odometry and SLAM: A Review. Chinese Journal of Aeronautics, 32(2), 287–299. https://doi.org/10.1016/j.cja.2018.08.003

[25] Chen, H., Jiang, Y., Liu, Y., & Wang, D. (2024). Evaluation and Benchmarking of Visual-Inertial SLAM Systems for Real-Time Applications. ISPRS International Journal of Geo-Information, 13(1), 163. https://doi.org/10.3390/ijgi13010163

[26] Chen, Y., Zhou, X., & Yuan, X. (2018). A Comparative Review of Visual-Inertial SLAM Algorithms in Mobile Robotics. Robotics, 11(1), 24. https://doi.org/10.3390/robotics11010024